

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平3-214233

⑬ Int. Cl.<sup>5</sup>

G 06 F 7/52  
11/22

識別記号

3 1 0 V  
3 3 0 G

庁内整理番号

7056-5B  
7230-5B

⑭ 公開 平成3年(1991)9月19日

審査請求 未請求 請求項の数 1 (全8頁)

⑮ 発明の名称 乗算器

⑯ 特 願 平2-10385

⑰ 出 願 平2(1990)1月18日

⑱ 発 明 者 崎 山 史 朗 大阪府門真市大字門真1006番地 松下電器産業株式会社内

⑲ 出 願 人 松下電器産業株式会社 大阪府門真市大字門真1006番地

⑳ 代 理 人 弁理士 山 本 孝

明 細 書

1. 発明の名称

乗 算 器

2. 特許請求の範囲

ウォーレスツリー方式の部分積変換を有する乗算器において、多段部分積を2段部分積に変換するまでのウォーレスツリー変換部を、部分積生成部によって生成された第1の多段部分積に複数個の加算器による演算を行って第2の多段部分積に変換する第1変換部と、この第1変換部により生成した第2の多段部分積に複数個の加算器による演算を行って第3の多段部分積に変換する第2変換部と、以下同様にして、第(n-1)変換部により生成した第nの多段部分積に複数個の加算器による演算を行って最終2段部分積に変換する第n変換部との、計n個の変換部に分けたとき、乗算器本体の入力ベクトルから前記各変換部毎の全ての全加算器または半加算器毎に同一の入力パターンを与え、かつ、前記入力ベクトルの種類により前記各変換部毎の全ての全加算器または半加算器毎

に任意の入力パターンを与えるテスト機構を設けたことを特徴とする乗算器。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、ディジタル信号処理LSI中で用いられる乗算器、特に故障検査ベクトルの簡易化を図ったウォーレスツリー(Wallace-tree)方式の乗算器に関するものである。

〔従来技術〕

一般に、ディジタル並列型乗算器の構成は、大要、部分積生成部と、この部分積生成部によって生成した多段部分積群を2段部分積群に変換する部分積変換部と、この部分積変換部によって生成した2段部分積群を加算する2入力加算部に分かれている。

このうち、前記部分積変換部の構成としては、一般にキャリーセーブ(carry-save)方式や、ウォーレスツリー方式のものが知られている。

キャリーセーブ方式の部分積変換部は全加算器が3入力2出力であり、全加算器で演算を行うと、

部分積が1つ減ることを利用し、多段部分積を上段から順次全加算器を用いて足し込んで行き、最終的に2段部分積に変換するものである。このキャリーセーブ方式はアレイ状であるため組み易く、また、故障検査ベクター（回路の全てのゲートのスイッチングを検査する入力ベクター）が作成し易い反面、桁上げ信号や和信号の伝搬ゲート数が最初の部分積生成部によって生じた多段部分積段数に比例して多くなるという不都合があり、高速性についてはウォーレスツリー方式よりも劣っている。

これに対して、ウォーレスツリー方式の部分積変換部はキャリーセーブ方式の多段部分積の足し込みを順次的に行うものではなく、並列して行うものであり、多段部分積の同じ位の部分積どうしを複数の全加算器を用いて同時に足し込み、それによって生じた桁上げ信号や和信号の多段部分積の同じ位どうしを、更に同時に複数の全加算器を用いて足し込むという操作を最終的に2段の部分積になるまで行うように構成されている。

（第2変換部）である。この部分積群の第2変換部(12)において、□印で示すものは第1変換部(11)の加算器(11-1-a)～(11-2-e)による演算を行った結果新たに生成された和信号、●印で示すものは桁上げ信号、○印で示すものは標準部分積群(11)で演算されなかった部分積群の要素信号がそのまま部分積群(12)の要素として与えられたものである。

また、(12-1-1)～(12-6-1)は部分積群(12)の各要素、(12-1-a)～(12-2-b)はそれぞれ加算器であって、前記部分積群(11)の場合と同様に短冊状長方形で示すと共に、斜線部分は半加算器を、白抜き部分は全加算器を示している。

(13)は部分積群(12)と同様に同部分積群の第2変換部による演算を行った結果、新たに生成された部分積群、(14)は前記各部分積群(11)(12)と同様に部分積群の変換部（第3変換部）(13)による演算を行った結果、新たに生成された部分積群である。

(15)はウォーレスツリー方式の部分積変換部に

このウォーレスツリー方式もキャリーセーブ方式と同様に、全加算器が3入力2出力であり、全加算器を1段通す毎に部分積が1つ減ることを利用し、最終2段の部分積に変換する方式であるが、桁上げ信号や和信号の伝搬がキャリーセーブ方式のように順次的に行うものではなく、並列に行われるから高速性の面では有利である。

第2図に8ビット×8ビットの乗算をウォーレスツリー方式により、2段部分積に変換する方法の一例を示す。この図において、(11)は8ビット×8ビットの乗算を行うとき部分積生成部により生成される標準部分積群のウォーレスツリー方式の第1変換部であり、図において○印で示す(11-1-1)～(11-8-8)はそれぞれ部分積群の要素、図において短冊状長方形で囲む部分で示す(11-1-a)～(11-2-e)はそれぞれ加算器であって、この内、斜線部分は半加算器を、白抜き部分は全加算器を示している。

(12)は部分積群の第1変換部(11)による演算を行った結果、新たに生成された部分積群の変換部

によって、第1変換部(11)の多段部分積群が2段の部分積群に変換された最終結果の部分積群である。ウォーレスツリー方式では例えば第1変換部(11)の多段部分積群の下から8の位の部分積では、全加算器(11-1-g)(11-2-c)によって同時に全加算され、同じ位の全加算器(12-1-f)の(12-1-8)(12-3-5)に和信号として、また1つ上の位の全加算器(12-1-g)の(12-2-7)(12-2-b)の(12-4-4)に桁上げ信号として伝搬される。

第2変換部(12)の部分積群の全加算器(12-1-f)(12-1-g)や半加算器(12-2-b)でも同様に同じ位の第1変換部(11)の演算部の和信号、桁上げ信号または演算されなかった標準部分積群の部分積信号で、全加算や半加算の演算がなされる。このような操作が全ての位で並列に実行され、最終2段の部分積群(15)に変換されるまで同様の操作が繰り返される。

8ビット×8ビットの乗算では、4段の全加算器の遅延時間（部分積群(11)～(14)の変換部の遅延時間）で最終2段の部分積群(15)に変換される。

これはキャリーセーブ方式では7段必要であったのに比べて高速性が大幅に向上している。

〔発明が解決しようとする課題〕

上述のように、ウォーレスツリー方式はキャリーセーブ方式と比較して高速性の面からは相当有利であるが、故障検査ベクターが非常に作成しにくいという問題点がある。

しかし、ウォーレスツリー方式の乗算器のテスト方式やテスト回路等のテスト手段は現在のところ、充分な実用性を備えた提案がなされておらず、このため、現状では乱数的な大量の故障検査ベクターにより故障検出率を上げたり、あるいは内部の故障検出の困難な加算器に強制的に外部から信号を与えるテスト機能を設けたりするような方法しかなく、効率良く故障検出率を高めるテスト方式は実現されていない。

ウォーレスツリー方式で $13 \times 12$ の乗算器を構成した場合、乱数により2000の検査ベクターを与えたとしても、回路故障検出率は60%にも満たないという結果も報告されている。

にはなる。

しかしながら、全てのベクターを実際に与えることは不可能で、現実には故障検出率が90%程度のベクターにとどめ、それで良品選出検査を行っているのが実情である。

本発明は、ウォーレスツリー方式を用いた乗算器における部分積変換部の故障検出率を簡単なテスト機構を付加することにより、数少ない故障検出ベクターで故障検出率を100%達成することが可能な乗算器の提供を目的とするものである。

〔課題を解決するための手段〕

上記目的を達成するために本発明の乗算器は、ウォーレスツリー方式の部分積変換を有する乗算器において、多段部分積を2段部分積に変換するまでのウォーレスツリー変換部を、部分積生成部によって生成された第1の多段部分積に複数個の加算器による演算を行って第2の多段部分積に変換する第1変換部と、この第1変換部により生成した第2の多段部分積に複数個の加算器による演算を行って第3の多段部分積に変換する第2変換

部に、一般に、全加算器や半加算器の故障検出ベクターを考えると、一つの全加算器への入力として、(ロウ、ロウ、ロウ)(ロウ、ロウ、ハイ)(ロウ、ハイ、ロウ)(ロウ、ハイ、ハイ)(ハイ、ロウ、ロウ)(ハイ、ロウ、ハイ)(ハイ、ハイ、ロウ)(ハイ、ハイ、ハイ)の8通りの入力ベクターを与えることで、初めて100%の故障検出が達せられ、半加算器では(ロウ、ロウ)(ロウ、ハイ)(ハイ、ロウ)(ハイ、ハイ)の4通りのベクターを与え、初めて100%の故障検出が達せられる。

いま、前述のウォーレスツリー方式を用いた8ビット×8ビットの乗算器の故障検出ベクターとして、例えば全てのベクター(2の16乗、つまり65536通り)を与えたとしても、全ての全加算器や半加算器に全ての入力パターンを与えることができるとは限らないため、故障検出率は100%になるとは限らない。尤も、実際問題としては、全てのベクターを与えれば、故障検出率は100%にならずとも良品選出検査としては完全

部と、以下同様にして、第(n-1)変換部により生成した第nの多段部分積に複数個の加算器による演算を行って最終2段部分積に変換する第n変換部との、計n個の変換部に分けたとき、乗算器本体の入力ベクターから前記各変換部毎の全ての全加算器または半加算器毎に同一の入力パターンを与え、かつ、前記入力ベクターの種類により前記各変換部毎の全ての全加算器または半加算器毎に任意の入力パターンを与えるテスト機構を設けたことを特徴とするものである。

〔作 用〕

本発明は上記構成により、テストモードにおいては、テスト機構によって第m変換部( $1 \leq m \leq n$ )の全ての全加算器の入力を同一とする入力ベクターを与えて、同入力ベクターを変化させることにより、第m変換部の全ての全加算器に全ての入力パターンを入力する。この全加算器の全ての入力パターンは前述したように8通りであるから、前記テスト機構により、第m変換部の全ての全加算器を故障検出するための入力ベクターの組は8

種類で実現でき、これにより第1～第n変換部からなるウォーレスツリー変換部の故障検出するための入力ベクターの組は $(8 \times n)$ 種類で実現できることになる。したがって、ウォーレスツリー変換方式を用いた乗算部の部分積変換部の故障検出率を、数少ない故障検出ベクターで100%達成することが可能になるものである。

#### 〔実施例〕

以下、本発明の実施例について図面を参照しながら詳細に説明する。第1図(A)は一般的なウォーレスツリー変換方式を用いた乗算器の構成図である。この図において、(1)は乗算器本体、(2)は乗算器本体(1)の部分積生成部、(3)はウォーレスツリー方式の変換部、(4)は2入力加算器である。

ウォーレスツリー変換部(3)は複数の変換部(31)～(3n)により構成される。(31)は部分積生成部によって生成された第1の多段部分積に複数の加算器による演算を行って第2の多段部分積に変換する第1変換部、(32)は第1変換部(31)により

生成した第2の多段部分積に複数の加算器による演算を行って第3の多段部分積に変換する第2変換部である。以下同様にして、(3m)は第(m-1)変換部(3n-1)により生成した第mの多段部分積に複数の加算器による演算を行い第(m+1)の多段部分積に変換する第m変換部(m:整数)、(3n)は第(n-1)変換部(3n-1)により生成した第nの多段部分積に複数の加算器による演算を行い最終2段部分積に変換する第n変換部(n:整数)である。

(A)(B)は乗算器(1)への入力ベクター、(C)は出力ベクターである。

(D)は通常の演算モードである実行モードと、故障検出のための演算モードであるテストモードの切換え信号である。

本発明は、ウォーレスツリー変換方式の乗算器におけるウォーレスツリー変換部の故障検出率を数少ない入力ベクター(A)(B)で実現するものであり、次のようなテスト方法をとる。

テストモードにおいては、第m変換部( $1 \leq m$

$\leq n$ )の全ての全加算器の入力を同一とする入力ベクター(A)(B)を与え、前記入力ベクター(A)(B)を変化させることにより、第m変換部の全ての全加算器に全ての入力パターンを入力する。

全加算器の全ての入力パターンは前述したように8通りであるから、もし上記方法を実現するテスト機構をもてば、第m変換部の全ての全加算器を故障検出するための入力ベクター(A)(B)の組は8種類で実現できる。つまり、第1～第n変換部からなるウォーレスツリー変換部の故障検出するための入力ベクター(A)(B)の組は $(8 \times n)$ 種類で実現できることになる。

第1図(B)は上記テスト機構を有する乗算器の具体的構成の一例を示し、この実施例では8ビット×8ビットのウォーレスツリー変換方式の乗算器に、本発明のテスト機構を付加した構成を備えている。

第1図(B)において、(31)は8ビット×8ビットの乗算を行うとき、部分積生成部により生成される標準部分積群のウォーレスツリー方式による

第1変換部であり、(1-1-1)～(1-8-8)の内の○印と×印は第1変換部(31)の部分積群の要素である。第1変換部(31)において(1-1-a)～(1-2-1)で示される短冊状長方形で囲む部分はそれぞれ全加算器である。(1-A)～(1-D)はテストモードと実行モードの切換え制御を兼ねた第1変換部(31)の部分積制御回路であって、これらの回路のうち、(1-A)(1-C)は全加算器で構成され、各部分積制御回路(1-A)～(1-D)への入力は以下に述べるように制御する。

すなわち、実行モード時には、部分積制御回路(1-A)～(1-D)の部分積要素はロウレベルの信号が強制的に与えられる。

テストモード時には第1変換部(31)の多段部分積群の同じ行にあるいずれか一つが入力される。ここでは、(1-1-1)が(1-1)に、(1-2-1)が(1-2)に、(1-3-2)が(1-3)に、(1-4-1)が(1-4)に、(1-5-1)が(1-5)に、(1-6-2)が(1-6)に、(1-7-3)が(1-7)に、(1-8-1)が(1-8)にそれぞれ入力されるものとする。

第1変換部(31)の多段部分積中の×印には、それぞれ同じ行の部分積制御回路の入力要素(1-1)～(1-8)の要素が入力されている。一例を挙げれば、(1-1-9)と(1-1-10)には(1-1)が入力しており、(1-6-1)と(1-6-10)には(1-6)が入力している。

(32)は第1変換部(31)による演算を行った結果、新たに生成された部分積群のウォーレスツリー方式第2変換部である。第2変換部(32)において、□印は加算器(1-1-a)～(1-2-j)による演算を行った結果、新たに生成された和信号である。○印は標準部分積群(31)で演算されなかった部分積群の要素信号がそのまま部分積群(32)の要素として与えられることを意味するものとする。(2-1-1)～(2-6-11)は部分積群(32)の各要素を示し、(2-1-a)～(2-2-k)と(2-A)(2-B)で示される短冊状長方形で囲まれた部分はそれぞれ全加算器を示している。このうち、(2-A)(2-B)は第2変換部(32)の部分積制御回路であって、部分積制御回路(2-A)の入力には(1-A)の和信号と桁上げ信号と(1-4)

の信号がそれぞれ(2-1)(2-2)(2-3)として与えられる。また、部分積制御回路(2-B)の入力には部分積制御回路(1-C)の和信号と桁上げ信号と(1-8)の信号がそれぞれ(2-4)(2-5)(2-6)として与えられる。

(2-1-a)～(2-1-i)の入力にはそれぞれ桁が対応する(1-1-a)～(1-1-i)の和信号と桁上げ信号、(1-4-1)～(1-4-8)の信号が入力される。一例を挙げると、全加算器(2-1-c)への入力は全加算器(1-1-d)の和信号と全加算器(1-1-c)の桁上げ信号と(1-4-2)の信号である。同様に、全加算器(2-2-a)～(2-2-k)の入力にもそれぞれ桁が対応する全加算器(1-2-a)～(1-2-j)の和信号と桁上げ信号および(1-8-1)～(1-8-8)の信号が入力される。第2変換部(32)の多段部分積中の×印にはそれぞれ同じ行の(2-1)～(2-6)の要素が入力される。一例を挙げると、(2-1-11)には(2-1)が入力しており、(2-6-1)(2-6-2)(2-6-3)には(2-6)が入力している。

(33)も第2変換部(32)と同様に、同変換部(32)

による演算を行った結果、新たに生成された部分積群のウォーレスツリー方式第3変換部で、内部構成や記号(□印や●印、○印、短冊状長方形で囲まれた部分など)の意味も第2変換部(32)と全く同様である。

部分積制御回路(3-A)の入力には部分積制御回路(2-A)の桁上げ信号と(2-B)の和信号と桁上げ信号がそれぞれ(3-2)(3-3)(3-4)として与えられる。全加算器(3-1-a)～(3-1-m)の入力にはそれぞれ桁が対応する全加算器(2-1-a)～(2-1-i)の桁上げ信号と、全加算器(2-2-a)～(2-2-k)の和信号と桁上げ信号が入力される。一例を挙げると、全加算器(3-1-c)への入力は全加算器(2-1-c)の桁上げ信号と、全加算器(2-2-b)の和信号と、全加算器(2-2-a)の桁上げ信号である。

第3変換部(33)の多段部分積中の×印は第1、第2変換部(31)(32)と同様にそれぞれ同じ行の(3-1)～(3-4)の要素が入力される。一例を挙げれば、(3-3-1)と(3-3-13)には(3-3)が入力しており、(3-4-1)と(3-4-2)には(3-4)が入力してい

る。(3-1)は部分積制御回路(2-A)の和信号が入力されているだけで、この実施例の場合、特に意味がない。しかし(3-1-1)～(3-1-11)の行の部分積がこの第3変換部(33)で加算器による変換が行われる場合には、(3-1)が×印への入力として使用される。

(34)も第2変換部(32)や第3変換部(33)と同様に、第3変換部(33)による演算を行った結果、新たに生成された部分積群のウォーレスツリー方式第4変換部で、内部構成や記号(□印や●印、○印、短冊状長方形で囲まれた部分など)の意味も第2、第3変換部(32)(33)と全く同様である。

部分積制御回路(4-A)の入力には(3-1)の信号と部分積制御回路(3-A)の和信号と桁上げ信号がそれぞれ(4-1)(4-2)(4-3)として与えられる。全加算器(4-1-a)～(4-1-m)の入力にはそれぞれ桁が対応する(3-1-1)～(3-1-11)の信号と、全加算器(3-1-a)～(3-1-m)の和信号と桁上げ信号が入力される。一例を挙げれば、全加算器(4-1-c)への入力は(3-1-6)の信号と、全加算器(3-1-c)の

## 特開平3-214233 (6)

和信号と、全加算器(3-1-b)の桁上げ信号である。

第4変換部(34)の多段部分積中の×印も第2、第3変換部(32)(33)と同様にそれぞれ同じ行の(4-1)～(4-3)の要素が入力される。

(35)はウォーレスツリー方式によって、多段部分積群(31)が2段の部分積群に変換された最終結果の部分積群であって、第1例(A)の2入力加算器(4)に対応するものである。

次に、この実施例の各動作モード(実行モード、テストモード)について説明する。

実行モード時は、第1変換部(31)の部分積制御回路(1-A)～(1-D)の入力要素((1-1)～(1-8))をロウレベルとするため、(2-1)～(2-6)、(3-1)～(3-4)、(4-1)～(4-3)の部分積制御回路の入力全てがロウレベルとなり、したがって、×印は全てロウレベルとなり、第1～第4変換部(31)～(34)で余分に付加した部分積要素は全て加算されなかったことになる。つまり、初めの標準部分積のみが最終結果の2段部分積群(35)に変換されたことになり、正しい演算結果を出力する。

～(1-1-1)が同じパターンを出力するため、全ての全加算器(2-1-a)～(2-1-1)の入力に同じパターンが与えられ、したがって、それらの出力も同じパターンとなる。

同様のことが第4変換部(34)でも行われる。

以上のように乗算器の入力ベクトルに①式のようなベクトルを与えると、テストモードでは各変換部毎の全ての全加算器毎に同一の入力パターンを与えることができ、かつ、①式の入力ベクトルの種類により前記各変換部毎の全ての全加算器毎に任意の入力パターンを与えることができる。

次に、ウォーレスツリー変換方式を用いた8ビット×8ビットの乗算部の部分積変換部の故障検査のベクトル数を求めてみる。

まず、全加算器の故障検査をするための検査ベクトル数は、発明が解決しようとする課題の項で述べたように8通りである。よって、第1変換部(31)では8通り(全加算器(1-1-a)～(1-1-1)と全加算器(1-2-a)～(1-2-1)の検査は同時に行える)、第2変換部(32)でも8通り(全加算器(2-1

次に、テストモード時の動作を説明する。テストモードでは前述のように、第1変換部(31)の部分積制御回路(1-A)～(1-D)の入力要素((1-1)～(1-8))には、(1-1-1)(1-2-1)(1-3-2)(1-4-1)(1-5-1)(1-6-2)(1-7-3)(1-8-1)が入力される。ここで、第1変換部(31)の部分積群の各行毎に等しくする乗算器の入力ベクトルを考えてみる。このベクトルは、

$$(オールハイ) \times (X) \cdots \cdots \text{①式}$$

を行えばよい。Xはどのようなベクトルでもよいことを表す。例えば8ビット×8ビットの乗算で、(1111111)×(01010101)というベクトルで演算を行うと、第1変換部(31)の部分積群の1,3,5,7の行がオールハイで、2,4,6,8の行がオールロウとなる。①式のようなベクトルを与えると、全ての全加算器(1-1-a)～(1-1-1)の入力に同じパターンが与えられ、したがって、それらの出力も同じパターンとなる。全加算器(1-2-a)～(1-2-k)も同様である。

また、第2変換部(32)でも、全加算器(1-1-a)

-a)～(2-1-1)と全加算器(2-2-a)～(2-2-k)の検査は同時に行える)、第3変換部(33)でも8通り、第4変換部(34)でも8通り、つまり合計32通りの乗算ベクトルを与えるだけで全ての全加算器の検査ができる。

但し、これらの中には全加算器の入力パターンに重なりがあるため(例えば第1変換部(31)の全加算器の入力に(ロウ、ロウ、ロウ)や(ハイ、ハイ、ハイ)のパターンを加えると、第2変換部(32)～第4変換部(34)の全ての全加算器の入力も(ロウ、ロウ、ロウ)や(ハイ、ハイ、ハイ)のパターンとなる)、実際は32ベクトルよりも可成り少なくなる。

### 〔発明の効果〕

以上説明したように、本発明の乗算器によるときは、ウォーレスツリー方式を用いた乗算部の多段部分積を2段部分積に変換するまでの部分積変換部に、乗算器本体の入力ベクトルから前記各変換部毎の全ての全加算器または半加算器毎に同一の入力パターンを与え、かつ、前記入力ベクトル

の種類により前記各変換部毎の全ての全加算器または半加算器毎に任意の入力パターンを与えるテスト機構を用いているので、テストモードにおいて、各変換部の全ての全加算器を故障検出するため入力ベクターの組は8種類で実現でき、これにより第1～第n変換部からなるウォーレスツリー変換部の故障検出するための入力ベクターの組は $(8 \times n)$ 種類で実現できる。

したがって、従来、乗数や被乗数のビット幅に対し、指数関数的に増加していた故障検出ベクターが数十のオーダーの故障検出ベクターで、しかも100%の故障検出ができるという優れた効果を発揮するものとなった。

また、このテスト機構を用いても、ゲート遅延段数は全く変わらず、また、回路の速度を犠牲にすることなく、テスト回路を実現することができるという、実用上の優れた効果を奏する。

なお、第1表に $n \times n$ ビットの乗算を行ったときのウォーレスツリー方式による全加算器の遅延段数と、本発明のテスト機構を用いたときの部分

積変換部の故障検査ベクターの数とを示す。

第1表

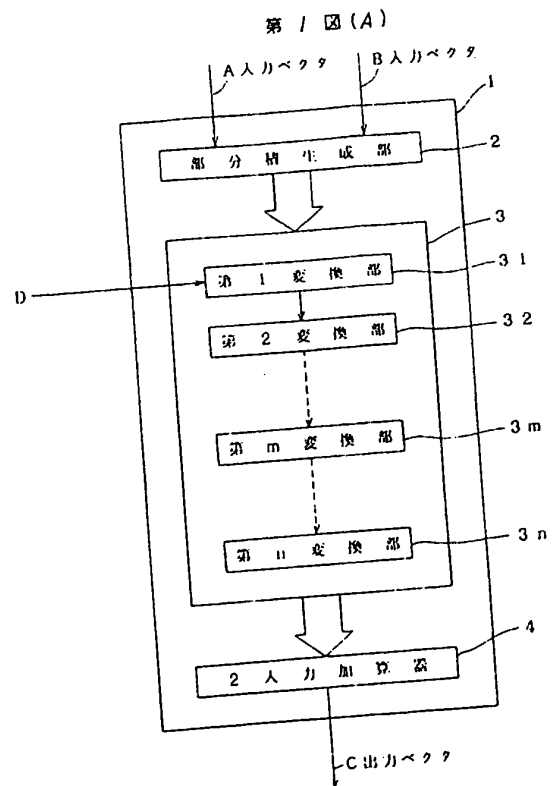
入力数	遅延段数	故障検査ベクター数
3	1	8
4	2	16
$5 \leq n \leq 6$	3	24
$7 \leq n \leq 9$	4	32
$10 \leq n \leq 13$	5	40
$14 \leq n \leq 19$	6	48
$20 \leq n \leq 28$	7	56
$29 \leq n \leq 42$	8	64
$43 \leq n \leq 63$	9	72

#### 4. 図面の簡単な説明

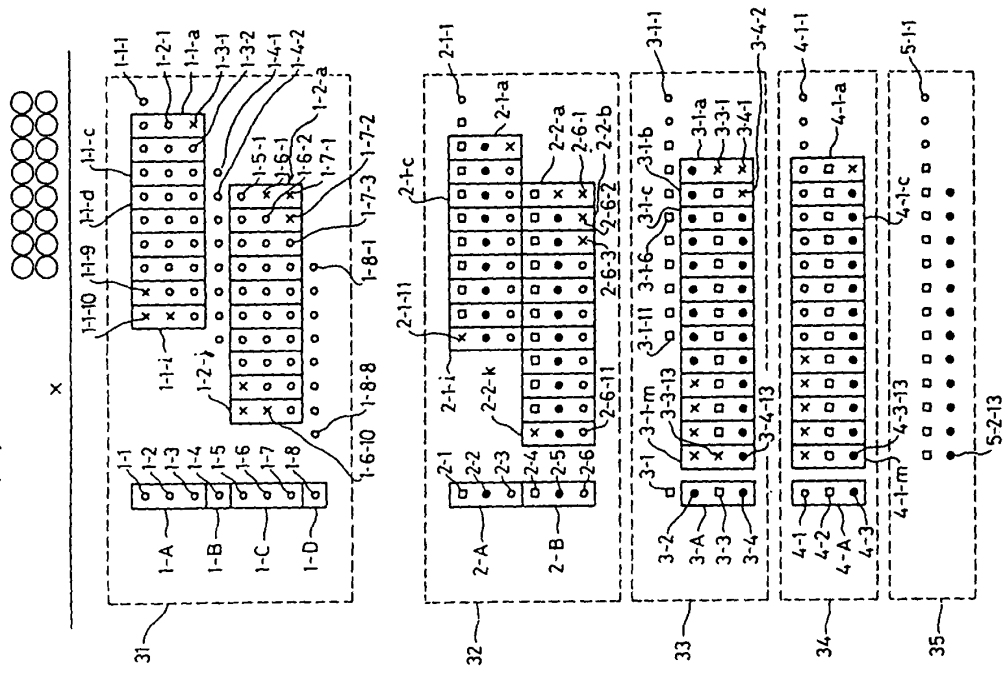
第1図(A)は本発明のウォーレスツリー変換方式を用いた乗算器の構成図、第1図(B)は本発明を8ビット×8ビットのウォーレスツリー変換方式乗算器に適用した実施例を示す構成図、第2図は従来の8ビット×8ビットのウォーレスツリー変換方式乗算器である。

- (1) …乗算器本体、(2) …部分積生成部、(3) …ウォーレスツリー変換部、(4) …2入力加算部、(31)～(3n)…変換部、(35)…2段部分積群、(1-1-1)～(1-8-8) …第1変換部分積変換部の部分積要素、(2-1-1)～(2-6-11)…第2変換部の部分積要素、(1-1-a)～(4-1-m) …全加算器、(1-A)～(4-D) …部分積制御回路、(1-1)～(4-3) …部分積制御回路の入力要素、(A)(B)…入力ベクター、(C) …出力ベクター、(D) …実行モードとテストモードの切換え信号。

特許出願人  
代理人弁理士 山 本



第1図(B)



第2図

